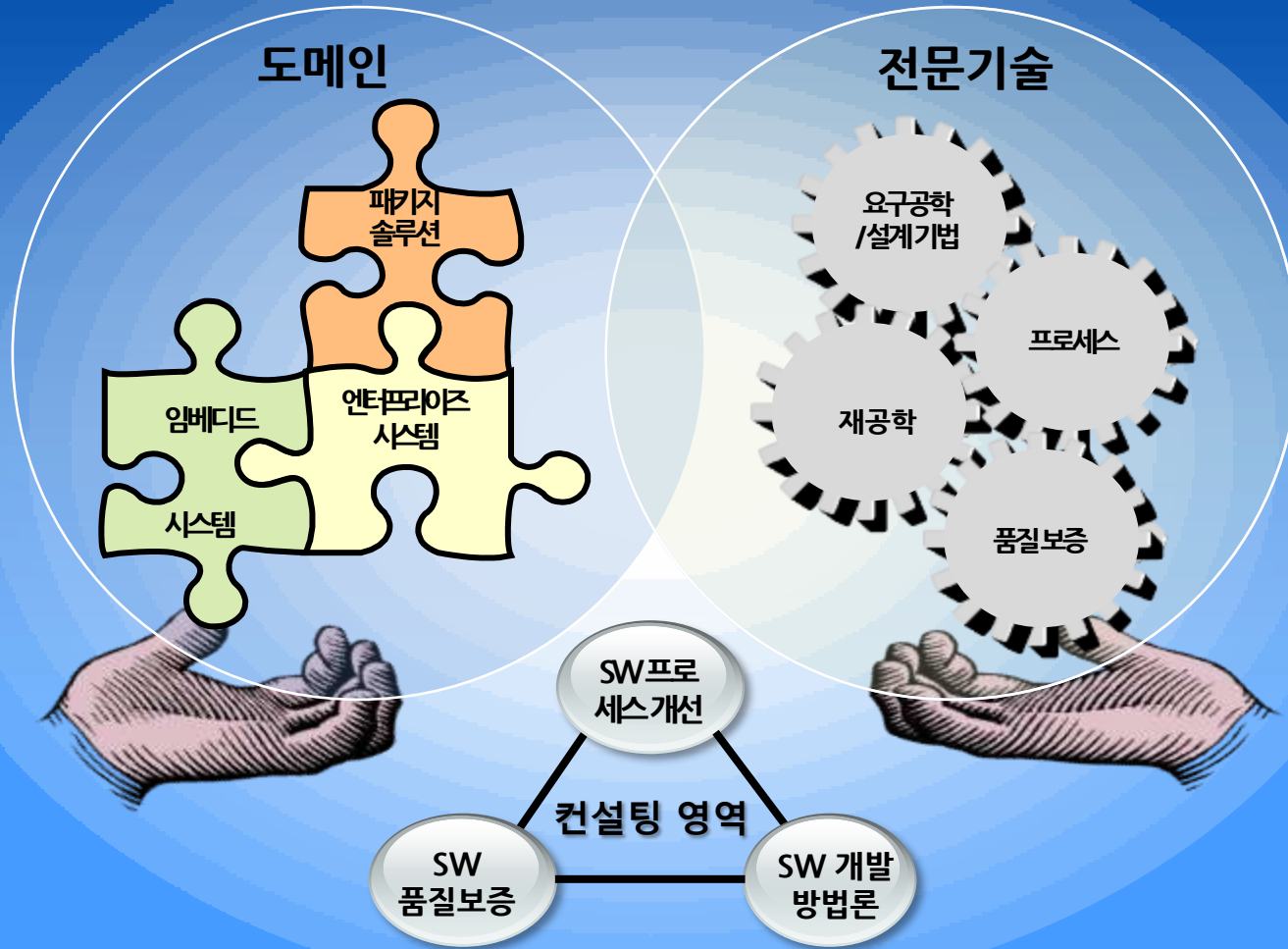


Clean Code

김진태

SEEG is ...



Clean Code 란?

- By 비야네 스트롭스트롭 (C++의 창시자)

- » 나는 우아하고 효율적인 코드를 좋아한다. 논리가 간단해야 버그가 숨어들지 못한다. 의존성을 최대한 줄여야 유지보수가 쉬워진다. 오류는 명백한 전략에 의거해 철저히 처리한다. 성능을 최적으로 유지해야 사람들이 원칙 없는 최적화로 코드를 망치려는 유혹에 빠지지 않는다. 클린 코드는 한 가지를 제대로 한다.

- By 그래디 부치 (Object Oriented Analysis and Design with Application의 저자)

- » 클린 코드는 단순하고 직접적이다. 클린 코드는 잘 쓴 문장처럼 읽힌다. 클린 코드는 결코 설계자의 의도를 숨기지 않는다. 오히려 명쾌한 추상화와 단순한 제어문으로 가득하다.

Clean Code 란?

- By 데이브 토마스 (이클립스 전략의 대부)

- » 클린코드는 작성자가 아닌 사람도 읽기 쉽고 고치기 쉽다. 의미 있는 이름이다. 특정한 목적을 달성하는 방법은 하나만 제공한다. 의존성은 최소이며 각 의존성을 명확히 정의한다. API는 명확하며 최소로 줄였다. 때로는 정보 전부를 코드만으로 명확하게 드러내기 어려우므로 언어에 따라 문학적 표현이 필요하다.

- By 존 제프리 (Extreme Programming Installed의 저자)

- » 중요한 순으로 나열하자면 간단한 코드는
 - 중복이 없다
 - 표현성이 뛰어나다
 - 작게 추상화되어 있다

Clean Code란?

컴퓨터가 이해하는 코드는 어느 바보나 다 짤 수 있다.

훌륭한 프로그래머는 사람이 이해할 수 있는 코드를 짤다

- 마틴파울러-

사람이 이해할 수 있는 코드란 무엇일까요?

- 가독성이 뛰어나다. *For people*
- 간단하고 작다. *For people*
- 의존성을 최대한 줄였다. *For people*
- 의도와 목적이 명확한 코드 *For people*
- 타인에 의해 변경이 용이 한 코드 *For people*
- 중복이 없는 코드 *For people*
- 개체(Class, Method)가 한가지 작업만 수행하는 코드 *For people*

왜 사람이 이해하는 코드를 만들어야 하나요 ?

SW 공장은 (**사람**)이기 때문입니다
일반적인 SW 개발 (**행태**)때문입니다

일반적인 SW 개발은 기존 코드에서 시작합니다

Enhanced

Modified

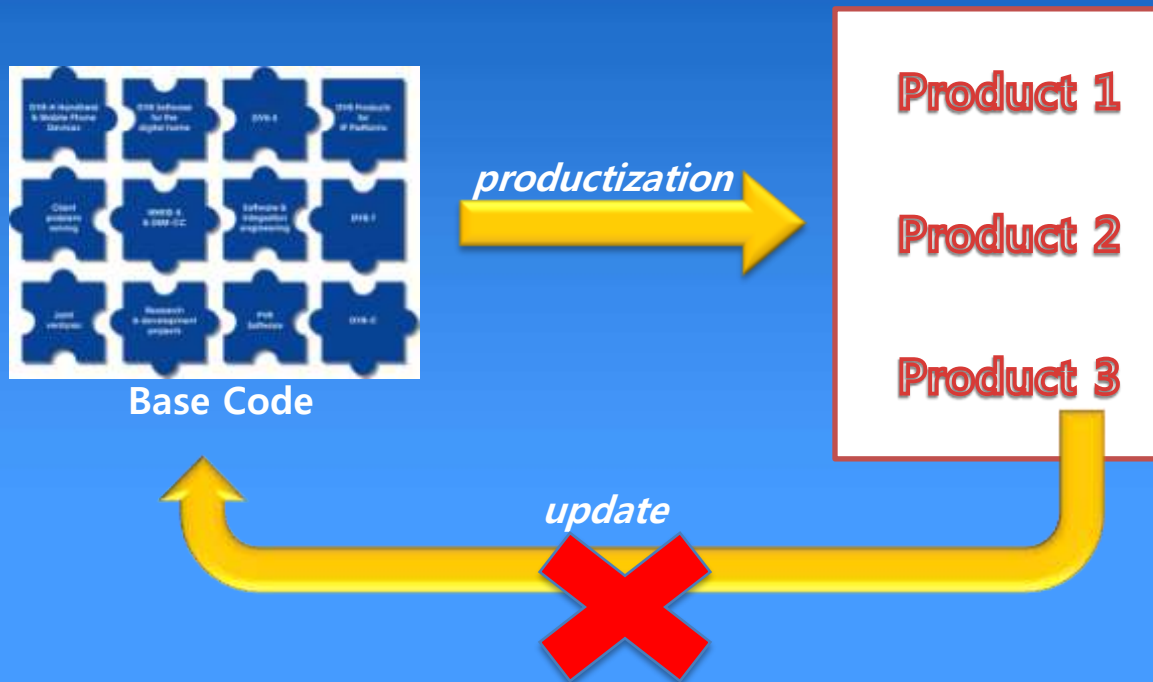
Adapted

Evolution

More complex

Drift away
from its original design

기능만 추가할 뿐 코드 품질의 개선 작업은 미비합니다



개선되지 않은 SW는 늪게 되어 Dirty code 가 됩니다



Dirty code는 SW Aging을 가속화
The Spiral of Complex

SW 노후화는 품질을 급격히 떨어뜨립니다

Symptom	Description	Metrics
Pollution	the system includes many components not necessary to carry out the business functions	Unused data, Duplicated code
No documents	the knowledge of the application domain and its evolution is spread over the programs and can no longer be derived from the documentation	Design spec, API doc, References
Poor lexicons	the names of components have little lexical meaning or are in any case inconsistent with the meaning of the components they identify	Naming convention
Tightly Coupling	the programs and their components are linked by an extensive network of data or control flows	Call graph, Fan-in, Fan-out
Architecture Erosion	the system's architecture consists of several different solutions that can no longer be distinguished; even though the software started out with a high quality basic architecture, the superimposition of these other hacked solutions during maintenance has damaged its quality	Dependency

왜 사람이 이해하는 코드를 만들어야 하나요 ?

- 코드가 건강합니다.
- 코드가 오래 갈 수 있습니다.
- 조직의 경쟁력을 높입니다.
- 개발 비용을 줄이고 생산성을 높입니다.

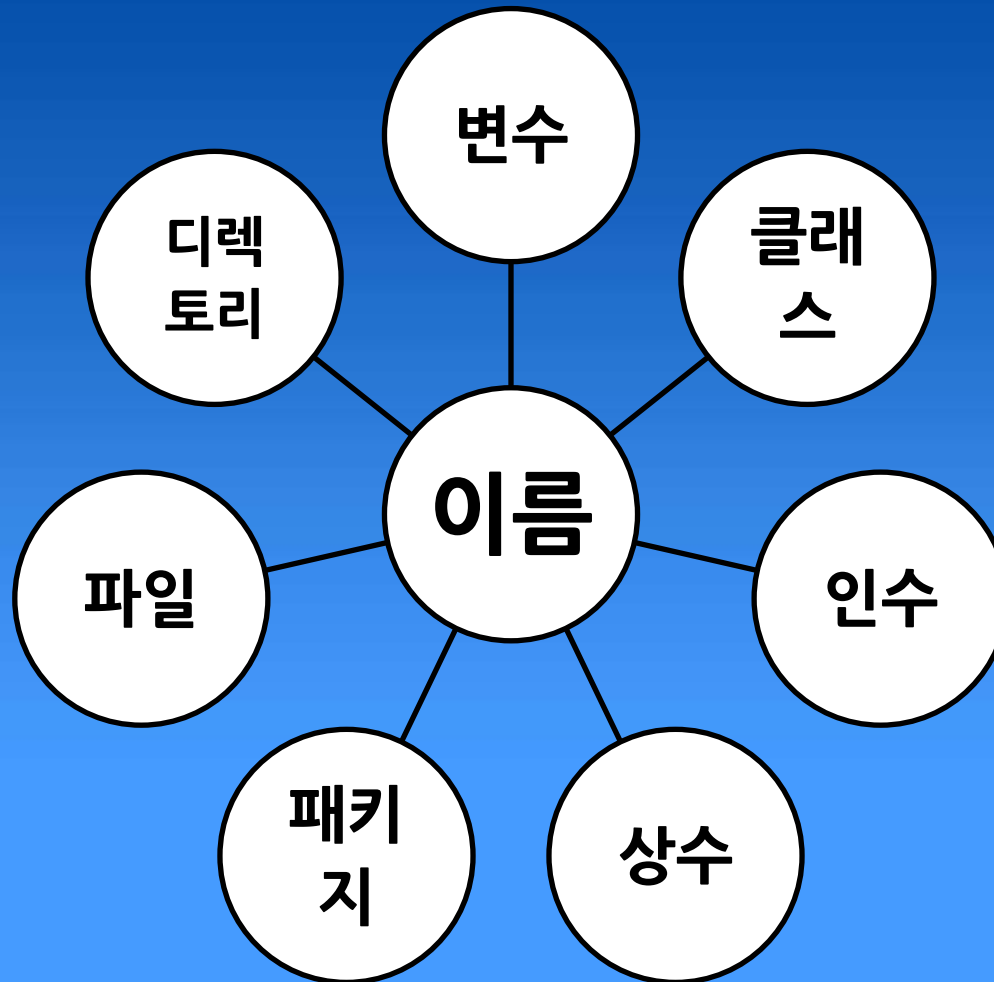
당신의 코드가 이렇게 되게 하십시오 (1)

의미 있는 이름

착한 함수

없는 게 나은 주석

이름 짓기가 가장 중요합니다.



의도를 분명히 하라

주석이 필요 없는 코드를 작성하라

```
//Login Client으로부터의 Login요청을 처리함.  
void CMessageProcFunc::Net_CLIENT_INFO()
```

```
public List<int[]> getList(){  
    List<int[]> list1 = new ArrayList<int []>();  
  
    for(int[] x : theList) {  
        if(x[0] == 4) {  
            list1.add(x);  
        }  
    }  
  
    return list1;  
}
```

```
void CMessageProcFunc::processLoginRequestFromClient()
```

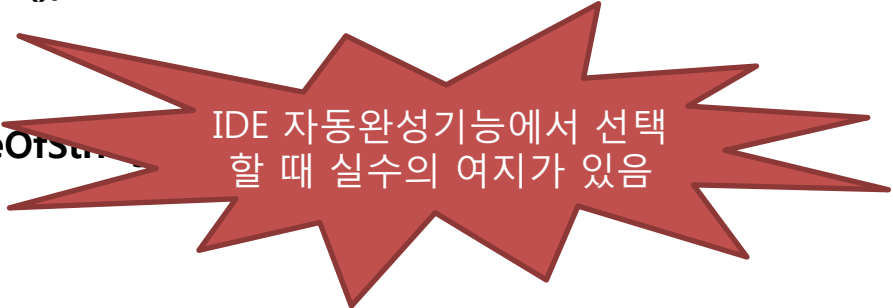
```
public List<Cell> getFlaggedCell(){  
    List<Cell> flaggedCells = new ArrayList<Cell>();  
  
    for(Cell cell : gameBoard){  
        if(cell.isFlagged()){  
            flaggedCells.add(cell);  
        }  
    }  
  
    return flaggedCell;  
}
```

그릇된 정보를 피하라

길고 서로 흡사한 이름은 피하라

```
public Class Vehicle {  
  
    controllForEfficientHandlingOfString ();  
    controllForEfficientStorageOfString ();  
  
}
```

```
public Class Y {  
    private Vehicle vehicle = new Vehicle();  
  
    public void method A(){  
        vehicle.controllForEfficientStorageOfString()  
    }  
}
```



IDE 자동완성기능에서 선택
할 때 실수의 여지가 있음

그릇된 정보를 피하라

연속적인 숫자를 붙인 이름을 피하라

```
public class Car{
    private Tire tire1 = new Tire();
    private Tire tire2 = new Tire();
    private Tire tire3 = new Tire();
    private Tire tire4 = new Tire();
}
```

```
public class Vehicle{

    public moveControl(int moveType){
        if(moveType==0){

        }else if(moveType==1){

        } else if(moveType==2){

        }else{
            throw Exception();
        }
    }
}
```

```
public class Car{
    private Tire leftFrontTire = new Tire();
    private Tire rightFrontTire = new Tire();
    private Tire leftBackTire = new Tire();
    private Tire rightBackTire = new Tire();
}
```

```
public class Vehicle{
    Const int GO_STRAIGHT = 0;
    Const int TURN_LEFT = 1;
    Const int TURN_RIGHT = 2;
    Const int STOP = 3;

    public controlMovement(int moveType){
        if(moveType== GO_STRAIGHT){

        }else if(moveType== TURN_LEFT){

        } else if(moveType== TURN_RIGHT){

        }else{
            throw Exception();
        }
    }
}
```

문맥에 맞는 단어를 사용하라

이해하는데 시간을 소모하지 않기 위해 일관성 있는 단어를 사용하라

```
public class Vehicle{  
  
    public double getFuelRemaining();  
    public double retrieveDegreeOfTireAbrasion();  
    public double fetchCurrentSpeed();  
  
}
```

```
public class Vehicle{  
  
    public double getFuelRemaining();  
    public double getDegreeOfTireAbrasion();  
    public double getCurrentSpeed();  
  
}
```

클래스 이름을 일관성 있게 작성하라

클래스이름은 명사 또는 명사구로 작성하라

: 클래스는 행위(멤버함수)의 주체로서 명사 또는 명사구로 표현됨

클래스	변경된 클래스 이름
CBandWidthControl	CBandWidth
CFileControl	CStationFile
CIOCPServer	CStationServer
CIOCPServerDlg	CStationServer
CQuotaCheckThread	CQuotaChecker
CSearchThread	CFileFinder
CServerControl	CStationServerController

함수 이름을 일관성 있게 작성하라

함수 이름은 동사 또는 동사구로 작성하라

: 함수는 클래스가 행하는 행위로서 동사 또는 동사구로 표현됨

SRP Test: *클래스가 스스로 멤버함수 한다.*

클래스	멤버함수
CIOCPServer	startService
CIOCPServerDlg	startStationServer
CIOCPServerDlg	makeLogPath
CIOCPServerDlg	writeLog
CMessageProcFunc	getSharedQuota
CMessageProcFunc	isAvailableToAddPhysicalDiskQuota
CMessageProcFunc	getUserPath
CMessageProcFunc	getAppendedUserPath
CMessageProcFunc	isAvailableToAddSharedQuota
CMessageProcFunc	getUserPath
CMessageProcFunc	increaseSharedQuota
CMessageProcFunc	parseDownloadFileList

단일책임의 원칙

SRP

단일책임원칙(Single Responsibility Principle)

- 클래스가 한가지 책임만 가져야 한다는 원칙
- 클래스나 모듈을 변경할 이유가 하나 뿐이어야 함
- 큰 클래스 → 작은 클래스 여럿
작은 클래스와 협력해 시스템에 필요한 동작 수행

_____가 스스로 _____ 한다.

각 줄의 첫 번째 공백에 모듈명을 적습니다. 두 번째 공백에 모듈의 메소드/함수 중 하나를 적습니다. 모듈의 모든 함수/메소드마다 이를 수행합니다.

각 줄을 소리 내어 읽습니다. 읽은 내용을 이해할 수 있습니까? 실제로 모듈의 함수/메소드가 의미하는 내용에 책임을 갖고 있어야 합니까?

SRP (Single-Responsibility Principle)

- SRP Test 예

- Automobile 모듈(혹은 클래스)에 SRP 적용하기

Automobile

```
start();  
stop();  
changeTires(tires []);  
drive();  
wash();  
checkOil();  
getOil();
```

SRP 준수



SRP 위반



착한 함수

작게 만들어라

이상적인 블록(if문 While문)안의 내용은 한 줄
블록 안에서 다른 함수를 호출하도록 작성

```
public String renderPageWithSetupsAndTeardowns(PageData pageData, boolean isSuite){

    If(isTestPage) {
        WikiPage testPage = pageData.getWikiPage();
        StringBuffer newPageContent = new StringBuffer();
        includeSetupPages(testPage, newPageContent, isSuite);
        newPageContent.append(pageData.getContent());
        ....
    }
    return pageData.getHTML();
}
```

```
public String renderPageWithSetupsAndTeardowns(PageData pageData, boolean isSuite){

    If(isTestPage) {
        includeSetupAndTeardownPages(pageData, isSuite);
    }
    return pageData.getHTML();
}
```


인수를 적게 만들어라

이상적인 인수개수는 0개이고 그 다음은 1개이다.
3개 이상의 인수가 필요하다면 객체 사용을 고려해라

```
public void drawCircle(double a, double b, double radius);
```

```
public void drawCircle(Point location, double radius);
```

```
public class Point{  
    double x;  
    double y;  
  
    double getX(){  
        return x;  
    }  
  
    double getY(){  
        return y;  
    }  
}
```

중복하지 마라

변경 시 여러 부분을 손대야 한다
오류가 발생할 확률이 높아진다

SearchThread 1162 ~

```
//QuataFree
stOut.ullQuataFree = 0;//youyou2_ULONGLONG

//File Name
_stprintf((LPTSTR)stOut.szName, TEXT("%s"), files.GetFileName());

//File Path
_stprintf((LPTSTR)stOut.szPath, TEXT("%s"), files.GetFilePath());

//File Path Rebuild Start
szFilePath.Format(_T("%s"),stOut.szPath);
nPoint = szFilePath.ReverseFind(_T('\\'));
szFilePath = szFilePath.Left(nPoint);
```

SearchThread 1314 ~

```
//QuataFree
stOut.ullQuataFree = 0;//youyou2_ULONGLONG

//File Name
_stprintf((LPTSTR)stOut.szName, TEXT("%s"), files.GetFileName());

//File Path
_stprintf((LPTSTR)stOut.szPath, TEXT("%s"), files.GetFilePath());

//File Path Rebuild Start
szFilePath.Format(_T("%s"),stOut.szPath);
nPoint = szFilePath.ReverseFind(_T('\\'));
szFilePath = szFilePath.Left(nPoint);
```



주식

주석대신 코드로 의도를 표현해라

코드만으로는 의도를 표현하기 힘들 때만 주석을 사용해라
그러나 될 수 있으면 코드로 표현해라

```
//플래그된 셀을 반환
public List<int[]> getList(){
    //플래그된 셀을 List형태로 저장
    List<int[]> list1 = new ArrayList<int []>();

    for(int[] x : theList) {
        if(x[0] == 4) { //플래그 되어있다면
            list1.add(x); // 리스트에 저장
        }
    }

    return list1;
}
```

```
public List<Cell> getFlaggedCell(){
    List<Cell> flaggedCells = new ArrayList<Cell>();

    for(Cell cell : gameBoard){
        if(cell.isFlagged()){
            flaggedCells.add(cell);
        }
    }

    return flaggedCell;
}
```

이런 주석은 삭제해야 합니다

없어도 되는 주석

```
/**
 * 달 중 날짜를 반환한다.
 *
 * @return 달 중 날짜
 */

Public int getDayOfMonth(){
    private int dayOfMonth; //달 중 날짜
    return dayOfMonth;
}
```

```
/** The name */
Private String name;

/** The version */
Private String version;

/** The licenceName */
Private String LicenceName;
```

이런 주석은 삭제해야 합니다

이력 및 저자를 기록하는 주석 SCM 도구를 사용해서 해결할 것

```
/*  
11-Oct-2001  
    작성자: 김민수  
    변경내용:  
    클래스를 다시 정리 하고 새로운 패키지 인  
    com.jrefinery.date로 옮김  
02-Nov-2001:  
    작성자: 홍길동  
    변경내용:  
    com.jrefinery.date로 옮김  
    NoitableDate 클래스 제거  
    getDescription() 메소드 추가  
*/
```

```
| hFile = CreateFile(strName, GENERIC_READ | GENERIC_WRITE, 0, NULL, OPEN_ALWAYS  
/*CREATE_ALWAYS*/, //youyou_200903_18_OPEN_ALWAYS FILE_ATTRIBUTE_NORMAL, NULL);
```

이런 주석은 삭제해야 합니다

주석 처리한 코드

```
CString strDir = _T("");
CString strName = _T("");
strDir = (LPTSTR)szDir;
strName = (LPTSTR)szName;

/*
if(strDir.GetLength() > PATHSIZE*2 || strName.GetLength() > PATHSIZE*2)
{
    TRACE(_T("[NetFILE_MAKEDIR]Path and Name is Too Long(CString)\n"));
    LogMessage(si->m_ChannelInfo->strId, "[NetFILE_MAKEDIR]Path and Name is Too Long(CString)\n");
    return;
}*/
```

```
else if(pRemove->cOListType == LIST_SHARE)
{
    /* //youyou_NoCheck_StorageSize_AtShare
    if(objFcFile.FileDir_IsExist(strFullTargetPath))
    {
        if(strFileType == _T("FLDR"))
        {
            ullSize = objFcFile.GetUsedDisk(strFullTargetPath); //[Byte]
        }
        else
        {
            objFcFile.GetFileLength(strFullTargetPath, &ullSize);
        }
        PlusShareUsedQuota(si, strShareIdentity, ullSize);
    }
    else
    {
        }
    }*/
}
//youyou_ShareQuota end
```

Clean Code, 왜 잘 안 되는가?

- 아는 것과 행하는 것은 별개의 문제
- SW 개발을 잘 하는 조직이란?
 - » 얼마나 많은 구성원이 얼마나 많은 Best Practice를 내재화하고 있는가의 문제

Clean Code 작성을 위해 조직이 해야 할 일

1. Do you use source control?
2. Can you make a build in one step?
3. Do you make daily builds?
4. Do you have a bug database?
5. Do you fix bugs before writing new code?
6. Do you have an up-to-date schedule?
7. Do you have a spec?
8. Do programmers have quiet working conditions?
9. Do you use the best tools money can buy?
10. Do you have testers?
11. Do new candidates write code during their interview?
12. Do you do hallway usability testing?

Clean Code에 대한 생각

- 정말 분석/설계가 문제입니까?
- 개발자는 소스코드와 함께 결함도 만들어 냅니다.
- 그런 개발자는 작업환경이 만들어 냅니다.
- 조직과 개인의 피와 땀이 없는 code의 고수는 없습니다.

Contact

- 김진태
- jtkim@swexpertgroup.com
- Twitter: [@swexpertgroup](https://twitter.com/swexpertgroup)